

1       1.     A method of protecting a selected region of an image from subsequent editing, the  
2     method comprising the steps of:

3               (a)     creating a first texture comprising a plurality of pixels each with an  
4     assigned scalar value indicating a level of protection for a corresponding pixel of a  
5     protected image;

6               (b)     directing graphical input into a second texture, wherein the protected  
7     image is at least initially unedited by the graphical input;

8               (c)     modifying a value of at least one pixel of the second texture using the first  
9     texture; and

10              (d)     blending at least one pixel of the second texture into the protected image.

1       2.     The method of claim 1, wherein the graphical input represents a plurality of brush  
2     strokes performed by a user.

1       3.     The method of claim 2, wherein the plurality of brush strokes comprises at least  
2     one overlapping portion.

1       4.     The method of claim 3, wherein the at least one overlapping portion corresponds  
2     to an area overlapped by a plurality of brush strokes.

1       5.     The method of claim 2, wherein the plurality of brush strokes comprises at least  
2     one member selected from the group consisting of a paint stroke, an erase stroke, a pencil  
3     stroke, a pen stroke, a line application, a character application, a text application, a batch  
4     deletion, a batch paste, and a flood fill.

1       6.     The method of claim 5, further comprising the step of generating the graphical  
2     input of step (b), wherein the graphical input corresponds to a movement of a user.

1      7.     The method of claim 6, wherein the step of generating the graphical input  
2     comprises assigning scalar values to pixels of a scratch texture that correspond to a  
3     transition region at one or more edges of a brush stroke.

1      8.     The method of claim 6, wherein the graphical input comprises a scratch texture  
2     representing a brush stroke and step (b) comprises blending the scratch texture into the  
3     second texture substantially upon completion of the brush stroke.

1      9.     The method of claim 8, wherein the step of blending the scratch texture into the  
2     second texture comprises performing a compositing operation.

1      10.    The method of claim 9, wherein the compositing operation is an overlay operation  
2     performed with pixels of A and B, where A comprises pixels having a paint color  
3     attenuated by the scratch texture and B comprises the second texture.

1      11.    The method of claim 6, wherein the step of generating the graphical input  
2     comprises the step of, for each of at least a plurality of pixels of a scratch texture:  
3                (i)     comparing a candidate scalar value from received data to an existing  
4     scalar value at a corresponding pixel of the scratch texture; and  
5                (ii)    assigning the candidate scalar value to the corresponding pixel of the  
6     scratch texture only if the candidate scalar value exceeds the existing scalar value.

1      12.    The method of claim 1, wherein the graphical input represents at least one brush  
2     stroke performed by a user, and wherein the at least one brush stroke comprises at least  
3     one overlapping portion corresponding to an area of a single brush stroke that overlaps  
4     itself.

1      13.    The method of claim 1, further comprising the steps of:  
2                (e)    copying at least one pixel of the protected image into a display image; and

3 (f) blending at least one pixel of the second texture into the display image.

1 14. The method of claim 13, wherein step (e) and step (f) are each performed prior to  
2 step (d).

1 15. The method of claim 13, wherein at least one of step (c) and step (f) proceeds  
2 pixel-by-pixel as the second texture accumulates graphical input.

1 16. The method of claim 13, wherein step (e) and step (f) are each performed prior to  
2 step (d), and wherein at least one of step (c) and step (f) proceeds pixel-by-pixel as the  
3 second stencil texture accumulates graphical input.

1 17. The method of claim 1, wherein the graphical input represents at least one erase  
2 stroke performed by a user.

1 18. The method of claim 17, further comprising the step of:

2 (e) modifying a value of at least one pixel in the protected image using the  
3 first texture.

1 19. The method of claim 18, wherein step (e) comprises attenuating a value of a pixel  
2 in the protected image subject to a minimum RGB $\alpha$  alpha value, where the minimum  
3 alpha value is determined using the first texture.

1 20. The method of claim 1, wherein the graphical input represents at least one paint  
2 stroke and at least one erase stroke performed by a user.

1 21. The method of claim 1, wherein step (c) comprises attenuating values of pixels of  
2 the second texture using values of corresponding pixels in the first texture.

1 22. The method of claim 1, wherein step (d) comprises performing a compositing  
2 operation.

- 1    23.    The method of claim 22, wherein the compositing operation is an overlay
- 2    operation performed with pixels of A and B, where A comprises the second texture and B
- 3    comprises the protected image.
- 1    24.    The method of claim 23, wherein A comprises the second texture as modified in
- 2    step (c).
- 1    25.    The method of claim 1, wherein step (c) and step (d) are performed substantially
- 2    simultaneously.
- 1    26.    The method of claim 1, wherein the assigned scalar value of a pixel in the first
- 2    texture indicates a level of protection from 0% to 100%.
- 1    27.    The method of claim 26, wherein the level of protection is a nonzero value less
- 2    than 100%.
- 1    28.    The method of claim 26, wherein the level of protection relates to an opacity.
- 1    29.    The method of claim 1, wherein the protected image is unedited by the graphical
- 2    input of step (b) until the blending in step (d).
- 1    30.    The method of claim 1, wherein step (b) is performed following a first user signal.
- 1    31.    The method of claim 30, wherein step (d) is performed following a second user
- 2    signal subsequent to the first user signal.
- 1    32.    The method of claim 31, wherein the graphical input in step (b) represents a
- 2    plurality of paint strokes performed by the user between the first user signal and the
- 3    second user signal.
- 1    33.    The method of claim 30, wherein the first user signal is a button click.
- 1    34.    The method of claim 1, wherein the first texture represents at least one user-
- 2    selected region of the image.

1       35.     A method of blending a brush stroke into a target image, the method comprising  
2     the steps of:

3               (a)     receiving data from a graphical user interface corresponding to a brush  
4     stroke;

5               (b)     for each of at least a plurality of pixels of a scratch texture:  
6                       (i)     comparing a candidate scalar value from the received data to an  
7     existing scalar value at a corresponding pixel of the scratch texture; and

8                       (ii)    assigning the candidate scalar value to the corresponding pixel of  
9     the scratch texture only if the candidate scalar value exceeds the existing scalar  
10    value; and

11               (c)     blending the scratch texture into a target image.

1       36.     The method of claim 35, wherein step (c) is performed substantially upon  
2     completion of the paint stroke.

1       37.     The method of claim 35, wherein the received data represent the brush stroke as a  
2     plurality of pillboxes.

1       38.     The method of claim 35, wherein the scratch texture comprises pixels  
2     corresponding to a transition region along at least one edge of the brush stroke.

1       39.     The method of claim 38, wherein a scalar value assigned to a pixel of the scratch  
2     texture within the transition region is a function of a distance of the pixel from an edge of  
3     the paint stroke.

1       40.     The method of claim 35, wherein step (c) comprises performing a compositing  
2     operation.

1    41. The method of claim 40, wherein the compositing operation is an overlay  
2    operation performed with pixels of A and B, where A comprises pixels having a paint  
3    color attenuated by the scratch texture and B comprises the target image.

1    42. A method of protecting a selected region of an image from subsequent editing, the  
2    method comprising the steps of:

3         (a) creating a first texture comprising a plurality of pixels each with an  
4    assigned scalar value indicating a level of protection for a corresponding pixel of a  
5    protected image;

6         (b) directing graphical input into a second texture, wherein the protected  
7    image is at least initially unedited by the graphical input, and wherein the graphical input  
8    represents a plurality of user brush strokes;

9         (c) copying at least one pixel of the protected image into a display image;

10        (d) modifying a value of at least one pixel of the second texture using the first  
11    texture;

12        (e) blending at least one pixel of the second texture into the display image;  
13    and  
14        (f) blending at least one pixel of the second texture into the protected image.

1    43. The method of claim 42, wherein step (b) is performed following a first user  
2    signal.

1    44. The method of claim 43, wherein step (f) is performed following a second user  
2    signal subsequent to the first user signal.

1      45.     The method of claim 44, wherein the graphical input in step (b) represents a  
2      plurality of paint strokes performed by the user between the first user signal and the  
3      second user signal.